EECS 3201 Digital Logic Laboratory

Lab5: Registers, Sequential Simulation and Measurement

Objective

The objective of this lab is to gain experience with counters and shift registers. We will use these in turn to gain experience in measuring relationships within your circuits using simulation and internal instrumentation.

Reference Material

Intel Signal Tap tutorials, available from the laboratory section of the course eClass page. Look over the tutorials and, if possible, attempt these before the lab (you need access to the board for SignalTap).

Part A Counters

You are provided with a Verilog counter file (counters.v) with 3 different counter implementations. Create a project and include the file **but do not modify the file/modules.**

1) In your top-level module create an instance of each of these counters and display the output on LEDR.
   - The counters should be 8-bits wide (again **do not modify counters.v**).
   - Add clock divider circuits (see lab 4) to produce a *fast clock* with a period of 50 counts of the 50 Mhz clock on the DE-10 lite and a *slow clock* with a period of 100 ms.
   - Select the clock for your counters based on SW[9]. Clear the counters individually with switch inputs.
   - Based on SW[1:0] select one of the 3 counter outputs to display at a time (e.g. for 2’b01 display the first counter, for 2’b10 show the second counter and for 2’b11 show the third counter).
   - Debug your circuit.
2) Use the RTL viewer (**Tools > Netlist Viewers > RTL Viewer**) to look at the circuits synthesized for each of the counter instances (click instances to see inside). Alternatively, you can right click on the instance directly in the **Hierarchy** tab and select **Locate Node > Locate in RTL Viewer.** For your report, export and image or screen shot of suitable views. Are the synthesized circuits as you expected? Explain. [if you are curious what the circuits look like after fitting you can select **Locate Node**

> Locate in Technology Map Viewer and see some of the optimizations ae performed].

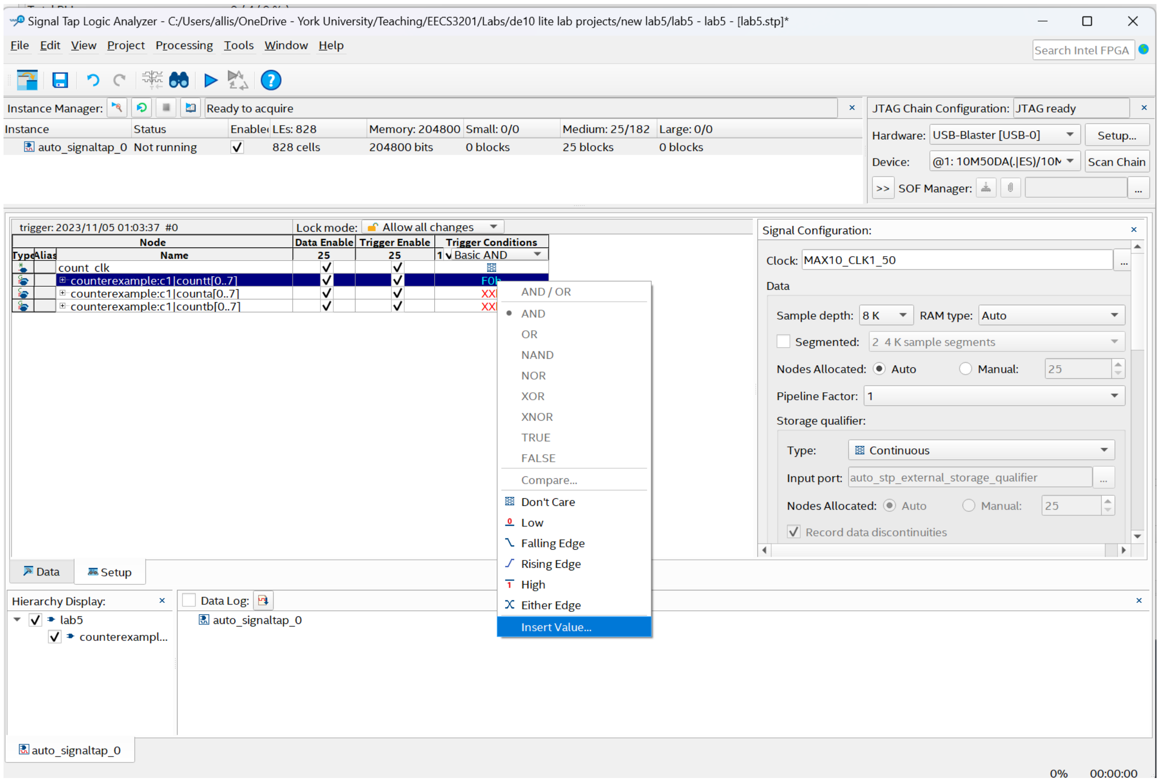
3) Simulate your designs (**Tools > Run Simulation Tool > RTL Simulation**). Make sure you include the outputs of each counter, the clock, and any internal signals you need (if internal signals get optimized away you can force Quartus to keep them with the compiler directive /**synthesis keep**/). Compare the simulations to the expected count behaviour. Do you see the expected differences between the implementations? For your report export images or screen shots of suitable views.

4) Use SignalTap to capture your circuit's behaviour under the fast clock. Note that SignalTap is a useful additional tool particularly because it can measure/respond to external signals but we will use internal signals.

   □ Using **File > New** choose **SignalTap Logic Analyzer File** and click OK. As explained in the tutorial add the signals including the clock for the counters, the 8-bit count for each counter. Group the counter outputs together by selecting them, right-clicking and selecting group.

   □ Use the MAX10_CLK1_50 signal (or the equivalent pin in the pin assignment) for the sampling clock (SignalTap clock, see tutorial). In that same tab set sample depth to 8k so that you get a long record.

   □ Set the trigger to capture when your asynchronous counter reaches a count of 8'h1F. To do his set the trigger as in the tutorial but for the group specify the trigger using *Insert Value.*



Signal Tap Logic Analyzer - C:/Users/allis/OneDrive - York University/Teaching/EECS3201/Labs/de10 lite lab projects/new lab5/lab5 - lab5 - [lab5.stp]*

Instance Manager: Ready to acquire

Instance	Status	Enable	LEs: 828	Memory: 204800	Small: 0/0	Medium: 25/182	Large: 0/0
auto_signaltap_0	Not running	✓	828 cells	204800 bits	0 blocks	25 blocks	0 blocks

JTAG Chain Configuration: JTAG ready

Hardware: USB-Blaster [USB-0]

Device: @1: 10M50DA(.|ES)/10M

SOF Manager:

trigger: 2023/11/05 01:03:37 #0

Lock mode: Allow all changes

Node			Data Enable	Trigger Enable	Trigger Conditions
Type	Alias	Name	25	25	1 ✓ Basic AND
		count_clk	✓	✓	▨
		⊞ counterexample:c1|countt[0..7]	✓	✓	F0h
		⊞ counterexample:c1|counta[0..7]	✓	✓	XX
		⊞ counterexample:c1|countb[0..7]	✓	✓	XX

Signal Configuration:

Clock: MAX10_CLK1_50

Data

Sample depth: 8 K RAM type: Auto

□ Segmented: 2 4 K sample segments

Nodes Allocated: ● Auto ○ Manual: 25

Pipeline Factor: 1

Storage qualifier:

Type: Continuous

Input port: auto_stp_external_storage_qualifier

Nodes Allocated: ● Auto ○ Manual: 25

☑ Record data discontinuities

Hierarchy Display:

☑ lab5

☑ counterexampl...

□ Data Log:

auto_signaltap_0

auto_signaltap_0

- □ Complete the remaining steps (check hardware, compile design, …) to configure SignalTap and capture the signals on the trigger. For your report export and image or screen shot of suitable views. Do you see the expected relationships between the counter outputs? Explain (hint, time scale of measurement is important here).

5) Demonstrate your 3 counters to the TA using both the fast and slow clocks. Demonstrate using the switches but be prepared to answer questions about your simulation or signal tap measurements.

Part B Linear feedback shift register circuit

Design a 5-bit linear feedback shift register circuit. The circuit is a standard 5-bit shift right register (similar to figures 5.17 and with feedback like 5.29 in the textbook) except that the input to the leftmost stage should be Q_2 ^ Q_0 rather than simply ~Q_0. Include an asynchronous preset and a parallel load capability in your circuit. The output of the circuit Q_0 is an unpredictable bit stream often used in test equipment and communications.

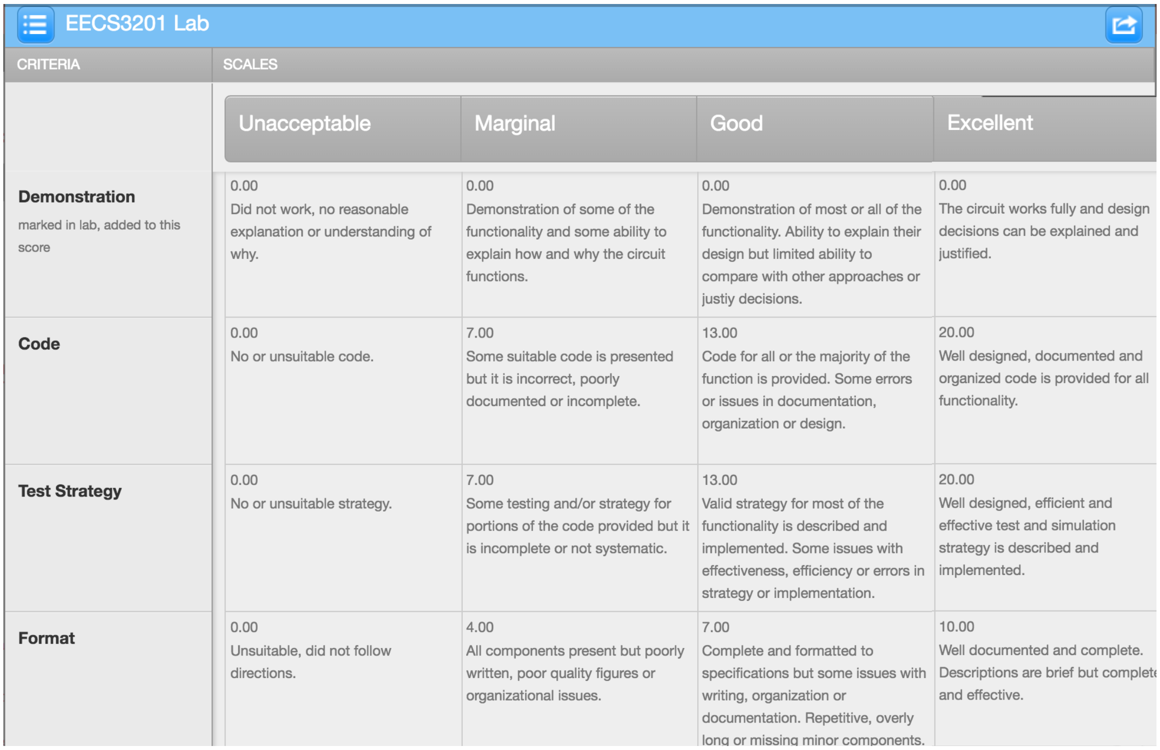
1) Download and debug your counter circuits using one of the pushbuttons as the clock input.
2) When you are satisfied that the shift, load and preset functions work change your clock to the fast clock from Part A.
3) Test the behaviour of your LFSR circuit using either ModelSim or SignalTap. Does the pattern ever repeat? If so, how often? Are there any problematic count states?
4) The implemented circuits must be demonstrated to the TA. You may demonstrate this part by showing the ModelSim or SignalTap outputs but be prepared to answer questions about your design.

Evaluation

Lab demonstration, in-lab explanations and answers, debug and test approach. [2 marks for fully working]

Lab report containing [graded out of 50, weight 3 marks of the overall lab grade]:

- Explanation of design approach and design including fully documented Verilog source including top level files (submit as .v files). For this lab the ‘code’ component includes proper configuration and implementation of the SignalTap and ModelSim files (can be documented with screenshots).
- Test patterns and/or a testing strategy. For this lab this will include the documentation and explanation of the ModelSim and SignalTap outputs and the answers to specific questions asked about the circuits.



CRITERIA	SCALES			
	Unacceptable	Marginal	Good	Excellent
Demonstration marked in lab, added to this score	0.00 Did not work, no reasonable explanation or understanding of why.	0.00 Demonstration of some of the functionality and some ability to explain how and why the circuit functions.	0.00 Demonstration of most or all of the functionality. Ability to explain their design but limited ability to compare with other approaches or justiy decisions.	0.00 The circuit works fully and design decisions can be explained and justified.
Code	0.00 No or unsuitable code.	7.00 Some suitable code is presented but it is incorrect, poorly documented or incomplete.	13.00 Code for all or the majority of the function is provided. Some errors or issues in documentation, organization or design.	20.00 Well designed, documented and organized code is provided for all functionality.
Test Strategy	0.00 No or unsuitable strategy.	7.00 Some testing and/or strategy for portions of the code provided but it is incomplete or not systematic.	13.00 Valid strategy for most of the functionality is described and implemented. Some issues with effectiveness, efficiency or errors in strategy or implementation.	20.00 Well designed, efficient and effective test and simulation strategy is described and implemented.
Format	0.00 Unsuitable, did not follow directions.	4.00 All components present but poorly written, poor quality figures or organizational issues.	7.00 Complete and formatted to specifications but some issues with writing, organization or documentation. Repetitive, overly long or missing minor components.	10.00 Well documented and complete. Descriptions are brief but complete and effective.